IPG Services

Warren Smith
Computer Sciences Corporation
NASA Ames Research Center

Introduction

- Goal: Location-independent computing
- Implementing a set of services to satisfy goal
- Build upon the Globus toolkit services
- Implementing with OGSA

Service Overview

- Credential Management
  - Storage of public and private keys
- Information Services
  - Distributed directory service
  - Event service
  - User Profile
- Grid Management
  - Monitoring and Testing: Ensure the operation of a grid
  - Problem Tracking
- Application Execution
  - Job Management: Reliable job execution
  - Portability Management: Tailor a job to specific systems
  - Resource Brokering: Intelligent resource selection
- Dynamic Accounting
  - "Accountless" access
  - Grid allocations
  - Resource pricing

Credential Management

- Storing public and private keys in ~/.globus can be a hassle
  - User may use > 1 machine as their home system
- MyProxy not very good for portals
  - User must log in to a Unix box before using portal
- Service that stores public and private keys for users
- User can contact service to get a proxy
  - Provide ID and passphrase
  - Provide proxy duration and limited or full
  - Retrieve a proxy
- Use this proxy as usual

Information Services

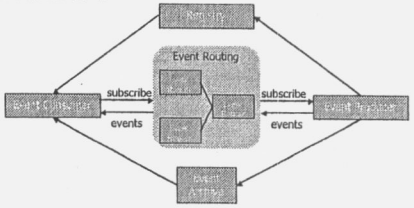
- Distributed directory service
- Event service
- User profile service

Distributed Directory Service

- Stores information about resources, services, applications, users, ...
- IPG is using the Globus MDS
- Analyze and improve performance of the MDS
  - IPG MDS has slow response times
- Extend information schemas
  - Scheduler, installed software, benchmarking
- Refine information schemas
  - Number of nodes != number of CPUs

Event Service

- Publish/subscribe model
- Archive Events
- Registry
  - Producers, consumers, archives, routing
- Existing system
- Layer atop OGSA
  - Currently using XML-based RPC
- Add functionality
  - Event routing
  - Event archives



User Profile Service

- Maintain a profile for each user
- Systems they prefer
- Environment they expect on each system
- Grid .cshrc

Grid Management

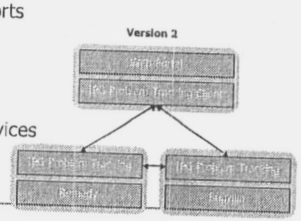
- Monitoring and testing service
  - Observe and report failures before users see them
- Problem tracking service
  - Interface local problem tracking services

Monitoring and Testing

- Observe and report failures before users see them
- Combination of active and passive tests

Problem Tracking

- A user finds a problem on a grid
- How does the problem get resolved?
  - Grid-wide support email lists
  - Local problem tracking systems
- Problem: Where to submit a trouble ticket?
- Interface to local problem tracking services
- Initial version
  - Interface to send problem reports
  - Web portal
- Next version may
  - Query problem status
  - Move tickets between local services



Dynamic Accounting

- Dynamic resource access
  - Use resources without previous per-user authentication
- Grid allocations
  - Charge resource use to remote allocations
- Resource pricing
  - Maintain prices for resource use
  - Set prices for resource use

Dynamic Resource Access

- Use resources without previous per-user authentication
  - E.g. Unix account
- Critical for scalable resource access
  - Don't want to apply for 100s of accounts
- Sites have full autonomy
- Sites can set user acceptance policy
  - "dynamic access is ok for Glenn users"
- Must be able to track all user access
  - Log mapping of grid ID to Unix username

Allocation Management

- Maintain user and group allocations
- Charge resource use to remote allocations
  - Run at Glenn, charge to Ames
- Allocation units
  - Standard units
  - Conversion between units
- Balance of payments
  - Center A uses Center B more than vice versa

Resource Pricing

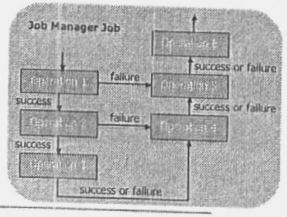
- Maintain prices for resource use
  - Owners set prices
  - Users query prices
- Investigate techniques to set prices
  - Agreement
    - Based on performance or uniqueness
  - Market-based

Application Execution Services

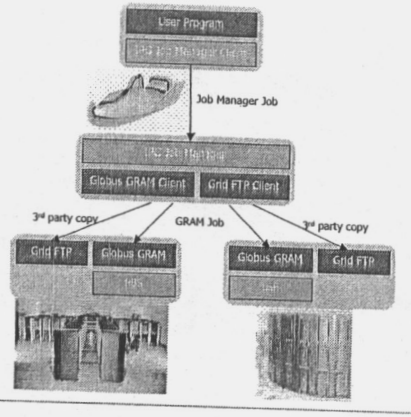
- Job management
  - Executing jobs on a grid
- Portability management
  - Making jobs portable
- Resource brokering
  - Selecting where to run

Job Management

- Reliable execution of a job
  - Job consists of operations
    - File operations: copy and delete files, copy and delete dirs
    - Execution operations
    - Specified exactly (hosts, paths, environment)
- Notification of job state change
- Maintain job database
- Designed to run a complex task (not a workflow)

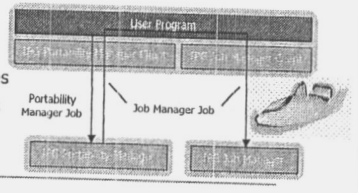


Job Management



Portability Management

- Help run jobs in a portable way
  - System-independent job -> system-dependent job
- Portability Manager job
  - Same structure as Job Manager job
  - System-independent
    - Don't have to set system-dependent characteristics
- Creates Job Manager job
  - System-dependent
  - Sets directories (home, scratch)
  - Picks correct executable
  - Sets environment variables (e.g. paths)
  - Adds files to stage for libraries
- Maintain software repositories
  - Executables and libraries
  - Data grid



Resource Brokering

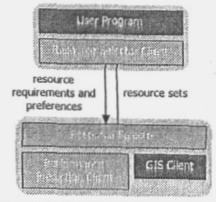
- Performance Prediction
  - Predict start times and execution times of applications
  - Predict file transfer times
- Resource Selector
  - Selects where to execute applications and where to obtain data
- Resource Broker
  - Executes a system-independent job
  - Uses Resource Selector, Portability Manager, Job Manager

Performance Prediction

- Predict execution time of applications
  - Use coarse information
    - User name, executable name, arguments, number of CPUs, ...
  - Historical database
  - Instance-based learning
    - Find similar experiences in database (distance function)
    - Form prediction from them (kernel regression)
- Predict scheduling queue wait times
  - Monitor scheduling queues
  - Simulate scheduling algorithms
- Predict file transfer times
  - Monitor network performance
  - Time series analysis

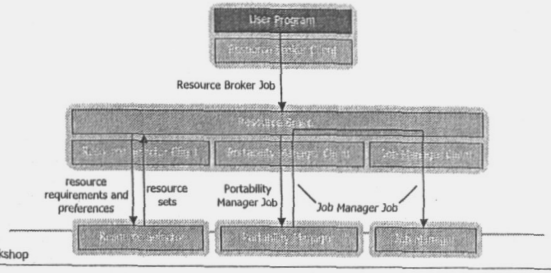
Resource Selector

- Selects resources for a job
- User specifies requirements and preferences
  - Resource characteristics
    - Operating systems, hosts, performance, cost, location
- Information obtained from
  - Grid information service
    - Resource characteristics
    - Accounting: Access, allocations, prices
  - Performance prediction services
- Returns a list of resource sets
  - Each job may require multiple resources
  - All resources meet user requirements
  - Sets are ranked by fitness



Resource Broker

- Executes a system-independent job
- Uses Resource Selector to select resources
- Uses Portability Manager to create system-dependent job
- Uses Job Manager to run job
- Redo process if job fails or executes slower than expected



Current Status

- Initial versions of
  - Event service
  - Job Manager
  - Resource Selector and Broker
  - Next versions in development
- Developing
  - Monitoring and Testing
  - Portability Manager
  - Performance prediction
  - Dynamic Accounting
  - MDS evaluation
- Availability
  - NASA only for now
  - Working on permission for wider releases